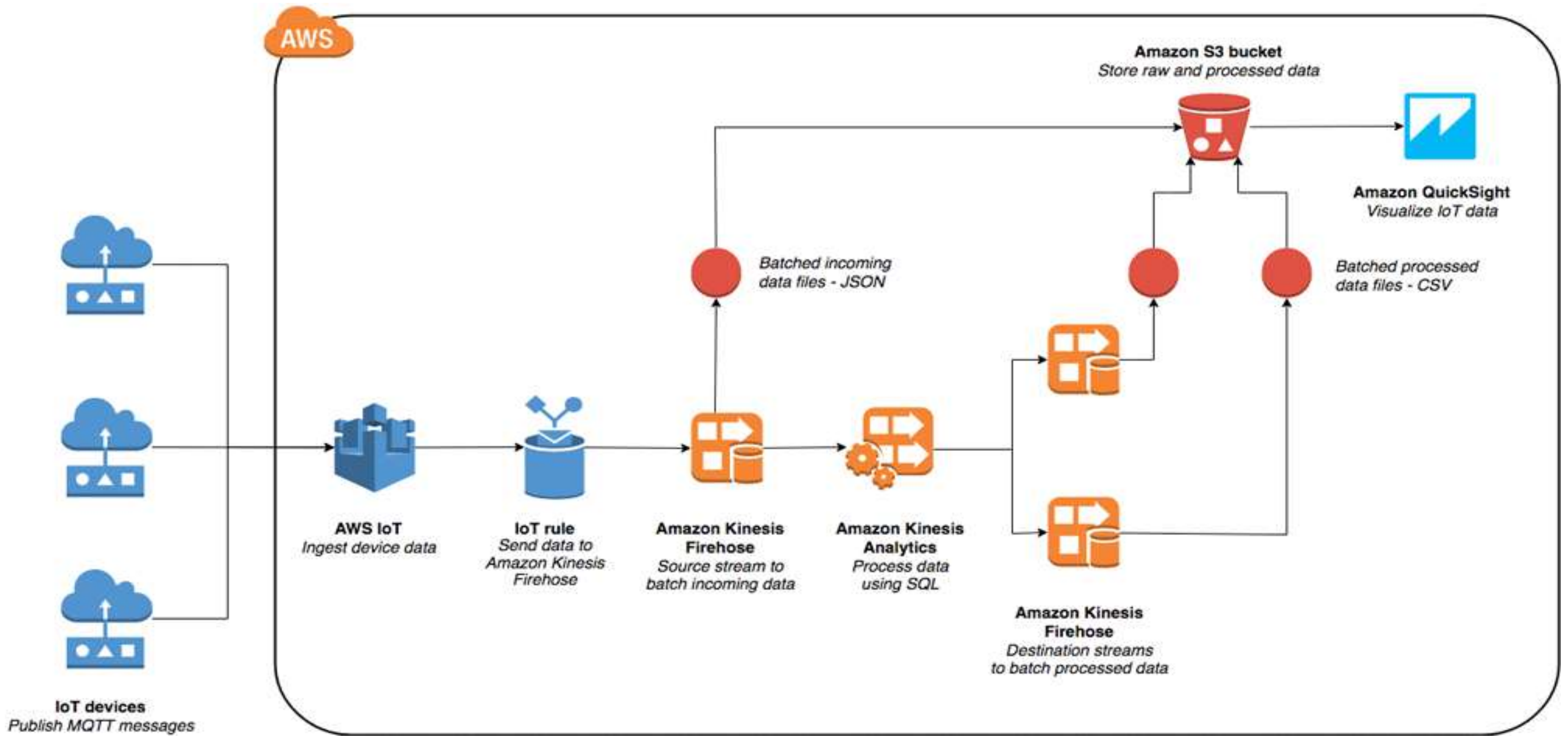
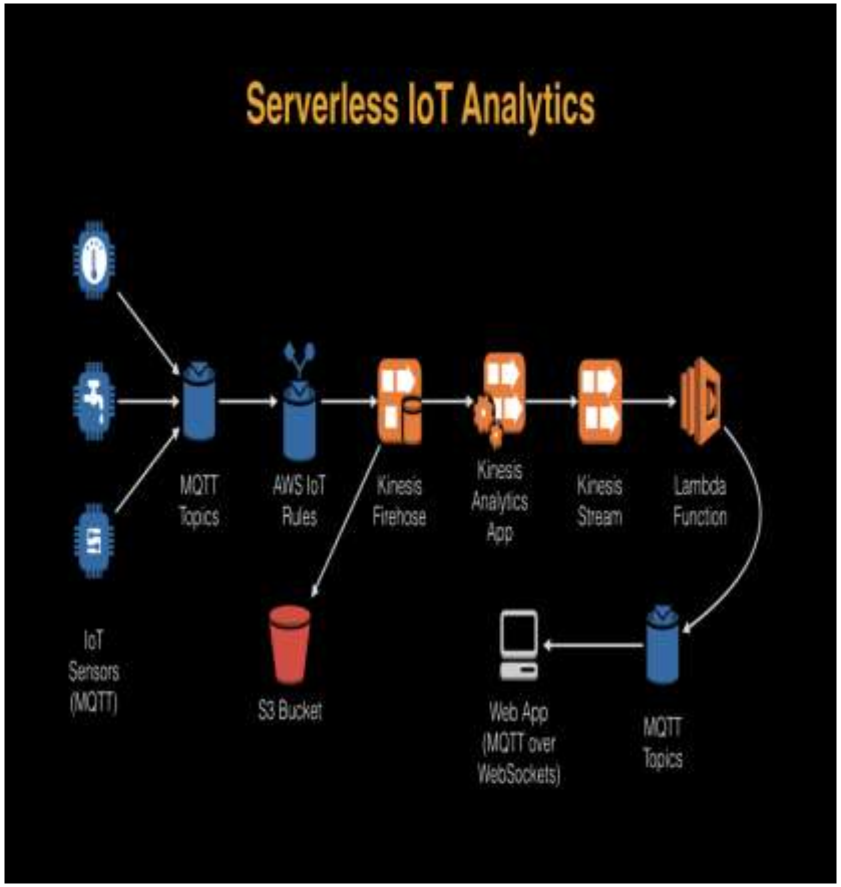
SERVERLESS IOT DATA PROCESSING


AWS
Amazon S3 bucket
Store raw and processed data
Amazon QuickSight
Visualize IoT data
Batched incoming data files - JSON
Batched processed data files - CSV
IoT devices
Publish MQTT messages
AWS IoT
Ingest device data
IoT rule
Send data to Amazon Kinesis Firehose
Amazon Kinesis Firehose
Source stream to batch incoming data
Amazon Kinesis Analytics
Process data using SQL
Amazon Kinesis Firehose
Destination streams to batch processed data

CONTENTS

ABSTRACT

When processing IoT data on a large scale, the cloud is no longer sufficient and it has been proposed to move parts of the computation closer to the IoT devices - the so-called fog computing. There are also three basic processing paradigms today that lend themselves to IoT data processing: stream and batch processing as well as serverless functions. Where to place which part of the data processing and which processing paradigm to choose, however, is often unclear.



INTRODUCTION

AI and cloud developmentAI is a branch of computer science that focuses on creating systems and software that can complete tasks that normally require human intelligence, such as decision making, natural language processing, learning, and reasoning. AI can be used to automate mundane tasks, like debugging and testing, using tools like AI-powered code assistants and bots. Additionally, AI can be used to enhance user experience and engagement with features like sentiment analysis, chatbots, voice assistants, and personalization with frameworks such as Azure Cognitive Services, TensorFlow, and PyTorch. Lastly, AI can be used to improve performance and efficiency by optimizing resource allocation, load balancing, and security through machine learning, deep learning, and neural networks.

IoT and cloud development

- IoT is a network of physical devices, sensors, and actuators that are connected to the internet and can exchange data and commands. This technology enables cloud developers to create applications and services that interact with the real world, such as building smart homes, cities, agriculture, and healthcare using platforms like AWS IoT, Google Cloud IoT, and Azure IoT. Additionally, IoT allows for collecting and analyzing large amounts of data like temperature, humidity, motion, and location using tools like Apache Kafka, Apache Spark, and MongoDB. Furthermore, it facilitates integrating and orchestrating multiple devices and services through protocols like MQTT, CoAP, and HTTP as well as frameworks like Node-RED, Apache NiFi, and Azure IoT Central

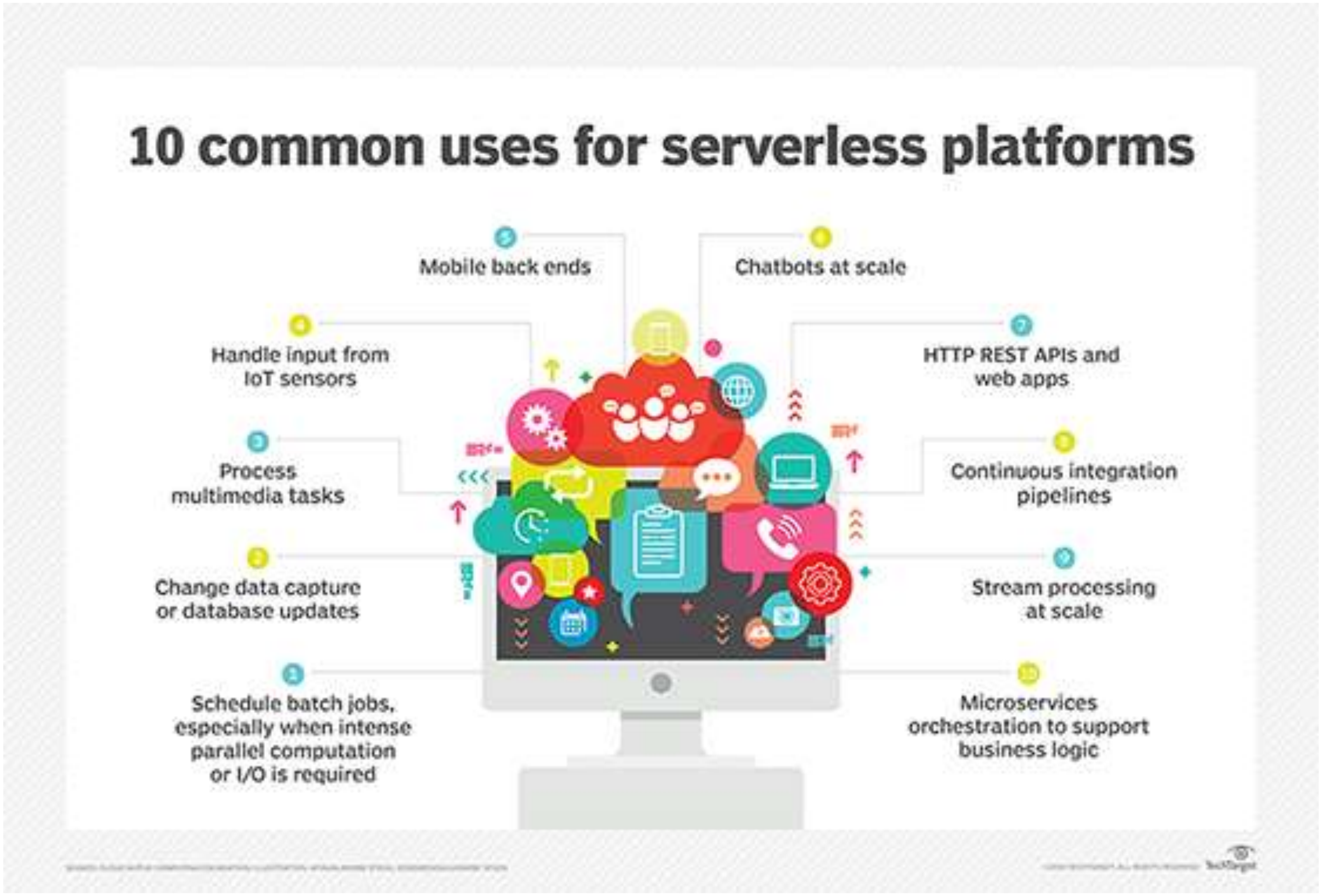
CHARACTERISTICS

The goal of event processing in an IoT context is to react to something that has happened in the physical world, in this case called an event For example, when an Internet-enabled button has been pressed, this triggers an event. There are a variety of reactions that can be appropriate for an event. Another IoT device could perform an action, a notification could be sent to an external system

CLOUD COMPUTING

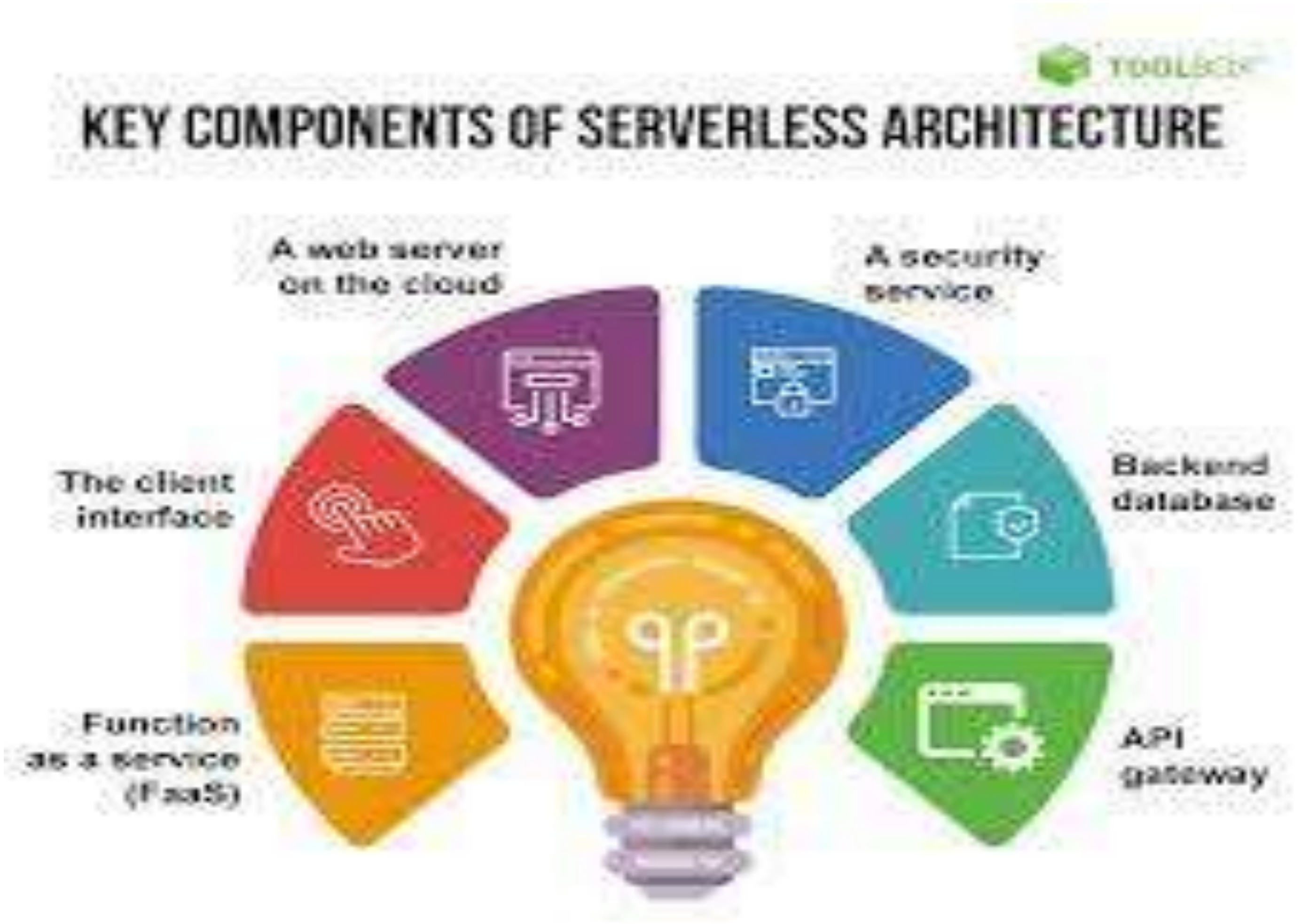
Serverless and cloud developmentServerless is a cloud development paradigm that abstracts away the underlying infrastructure, allowing developers to focus on the business logic and functionality of their applications and services. This can help developers in numerous ways, such as reducing operational overhead and complexity by using services like AWS Lambda, Google Cloud Functions, and Azure Functions. Serverless also increases agility and productivity through tools like Serverless Framework, AWS SAM, and Google Cloud Run. Additionally, it lowers costs and improves scalability with models like pay-per-use and pay-per-request

10 COMMON USES FOR SERVERLESS PLATFORM



10 common uses for serverless platforms
1 Schedule batch jobs, especially when intense parallel computation or I/O is required
2 Change data capture or database updates
3 Process multimedia tasks
4 Handle input from IoT sensors
5 Mobile back ends
6 Chatbots at scale
7 HTTP REST APIs and web apps
8 Continuous integration pipelines
9 Stream processing at scale
10 Microservices orchestration to support business logic
TechTarget

KEY COMPONENTS OF SERVERLESS ARCHITECTURE


KEY COMPONENTS OF SERVERLESS ARCHITECTURE
A web server on the cloud
A security service
The client interface
Backend database
Function as a service (FaaS)
API gateway

SOURCE CODE

- Creating source code for serverless IoT data processing typically involves using cloud services like AWS Lambda, Azure Functions, or Google Cloud Functions. Below is a simplified example using AWS Lambda in Python to process IoT data. This example assumes you're receiving data from IoT devices and want to process it.

CODE

```
import json
 def process_iot_data(event, context):
   # Parse the IoT data from the event    iot_data = json.loads(event['body'])
   # Perform data processing, e.g., filtering, transformation, or analysis
     processed_data = process_data(iot_data)
   # Store or send the processed data to your desired endpoint or service
     send_processed_data(processed_data)
   # Return a response to acknowledge processing
     response = {        "statusCode": 200,        "body": json.dumps({"message":
     "IoT data processed successfully"})
  }
     return response
   def process_data(iot_data):
   # Your data processing logic here
     processed_data = iot_data
   # In this simple example, we pass data as is    return processed_data
   def send_processed_data(data):
  # Implement logic to send data to your storage or other services    pass
```

CONCLUSION

- In this work, we have presented a framework for deciding when and where to implement functions, stream processing, and batch processing in an IoT data processing context. For this purpose, we have discussed the different use-cases of event processing and data analytics, highlighting the capacities of on-device, edge, fog, and cloud computing and demonstrating the strengths and weaknesses of each approach. Of course, IoT is a wide topic and we are not able to cover all possible data processing use-cases. We have also limited our approach to a hierarchical network of devices, edge gateways, fog nodes, and the cloud, while in reality there may be more or less components, or they may be connected differently.

THANK YOU